2007－ARC－174 (5)
2007／8／1

TLBを用いるキャッシュ利用状況推定の高精度化

小川周吾[†,††] 菅原豊[†] 平木敬[†]

近年多くのマルチコア，マルチスレッドプロセッサ上では共有キャッシュを持ち，スレッド間でキャッシュの競合が発生する．スレッド毎のキャッシュ使用状況が分かれば事前に競合頻度の推定，低減が可能である．しかし共有キャッシュに対して，キャッシュミス回数の情報を基にした使用状況の推定方式は利用できない．本稿ではキャッシュ使用状況の予測手段として，ページのアクセス頻度推定方式を提案する．我々はアクセス頻度推定方式をシミュレータ上でSPEC CINT2000を用いて評価した．その結果実際にアクセス頻度の高いページがアクセス頻度が高いと判定されることを示し，アクセス頻度の高いページの特定に有用であることが判明した．

An Improvement of Cache Usage Estimation with TLB Information

Shugo Ogawa,[†,††] Yutaka Sugawara[†] and Kei Hiraki[†]

Many multicore and multithread processors have a shared cache, therefore cache conflict is increased between executing threads. We can estimate and decrease cache conflict with cache usage condition. However, we cannot use the number of cache misses in estimating cache conflict if caches are shared between threads on processors. In this paper, we propose an estimation methods that estimates access frequency of pages for CPU cache usage condition. We evaluate our new methods with SPEC CINT2000 on a simulator, and we show that frequently accessed pages have an estimation that are accessed frequently. As a result, we show the fact that we can find pages that are accessed frequently with our methods.

1. 序　論

近年共有キャッシュを持つマルチコア，マルチスレッドプロセッサが普及している．キャッシュは同時実行中の複数スレッド間で共有されるため，スレッド同士の干渉によるキャッシュ競合が発生する．キャッシュの競合ミスはキャッシュの連想度を超えた同一領域へのデータ格納時に発生する．共有キャッシュは複数スレッドから使用されるため，単一スレッドとは異なる原因で競合が生じる．例えば複数スレッドがキャッシュの同一ラインを同時使用した場合に各スレッドでのキャッシュ使用量の合計が連想度を超えて，スレッド単体では発生しないキャッシュ競合が生じる．

複数スレッド間で生じる競合の頻度はキャッシュ使用状況から推定可能である．競合頻度の推定結果は競合の低減に使用可能である．キャッシュ競合頻度を調べる確実な方法は，キャッシュミス回数の計測である．キャッシュミス回数は，Page Coloring[3)4)]においてキャッシュ競合の検出，回避を動的に行うために利用される．キャッシュミス回数を用いる利点は競合頻度を直接得られる点である．得られるミス回数の履歴の作成により競合頻度を予測可能である．

しかし共有キャッシュではキャッシュミス回数による競合頻度の予測が困難である．それは共有キャッシュのキャッシュミス頻度は実行中の他スレッドのキャッシュ使用状況に影響されることが原因である．よってキャッシュミス回数を使用しない予測手段が必要である．

キャッシュ使用状況の予測手段として，本稿ではスレッドのアクセスしたページに対するアクセス頻度推定方式を提案する．スレッドのアクセスページからアクセス頻度の高いページを検出する．キャッシュに格納されるページはアクセス頻度の高いページであるため，検出したページからキャッシュ使用状況を推定可能である．提案方式によりアクセス頻度の高いページが検出されることを評価で示す．

2. 関連研究

本章ではキャッシュ競合頻度の推定，回避の関連研究としてPage Coloringについて説明する．またマル

† 東京大学大学院情報理工学系研究科
The University of Tokyo, Graduate School of Information Science and Technology
†† NEC システムプラットフォーム研究所
NEC System Platform Research Laboratories

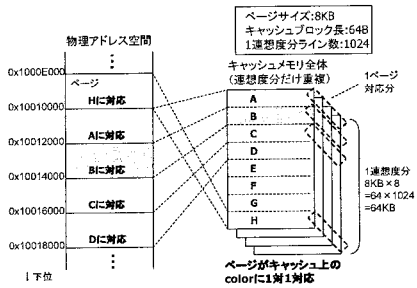

図 1　ページアドレスと color の関係

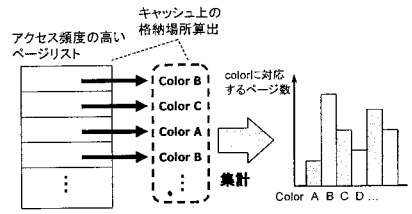

図 2　各 color に格納されるアクセス頻度の高いページ

チコア，マルチスレッドプロセッサ上での共有キャッシュにおけるミスの検出，回避方式について説明する．

2.1 Page Coloring

Page Coloring[1)2)] はアクセスするページのキャッシュ上の位置の偏りを防ぐメモリ確保方式である．メモリ上の各ページはキャッシュをページサイズ毎に区切った領域である一定の color に格納される (図 1)．

ページアドレスを調べることで，各 color に格納されるページ数が判明する．各 color のページ数を均等にすることでキャッシュの競合ミスを削減する．

これらの静的な Page Coloring に対して Bershad ら[3)4)] は Page Coloring を動的に行う方式を提案している．キャッシュミス回数を調べることで競合の発生している color を動的に検出する．その color に属するページを recolor，つまり別 color になるように移動してキャッシュ競合を回避する．この方式は単一スレッドによるキャッシュ占有を前提とする．そのため共有キャッシュでは他スレッドの影響によりキャッシュ競合頻度の推定，回避方式としての使用は困難である．

2.2 共有キャッシュにおけるミスの検出，回避方式

マルチコア，マルチスレッドプロセッサ上の共有キャッシュでは実行中の複数スレッド間で競合が発生する．我々は文献[7)] で，SMT プロセッサ上の共有キャッシュ上のミス削減方式を提案した．この方式ではプロセッサの性能モニタリングカウンタを用いて各スレッドのキャッシュミス回数を取得する．この情報からミスの増加を回避するように同時実行スレッドを選択する．

共有キャッシュのミス回数の履歴を同時実行されたスレッドの組み合わせ単位で記録する．各スレッド単位で履歴を作成しないのは，前述の通り共有キャッシュでは競合頻度に他のスレッドが影響するからである．プロセッサのスレッド数が N，システムのプロセス数が n の場合，履歴総数は n^N である．よってプロセッサのスレッド数が増加した場合に履歴総数が指数的に増大するため適用可能なシステム規模が限定される．

3. ページアクセス頻度推定方式の提案

本稿ではキャッシュ使用状況の予測手段として，ページアクセス頻度推定方式を提案する．提案方式の説明に先立って，提案方式を用いたキャッシュ使用状況の推定手順を説明する．まず Page Coloring と同様にキャッシュをページサイズ単位に区切った領域 (color) に分ける．次に提案方式により，各ページのアクセス頻度を求める．最後に各 color に格納される，アクセス頻度の高いページ数を求める (図 2)．以上の手順から各 color でのキャッシュ使用状況を推定する．以下では提案方式を更に以下の 2 段階に分けて説明する．

- アクセスページの検出
- 各ページのアクセス頻度の推定

3.1 アクセスページの検出方式

提案方式の前半の処理は，アクセスページの一覧作成を目的とする．アクセスページの一覧を作成するためには以下の 2 種類の処理が必要である．

- ページへのアクセス有無の検出
- アクセスページの集計

まずアクセスページの集計及び集計時期について説明する．集計ではアクセスページの検出情報からアクセスページの一覧を作成する．一覧作成をまとめて行う時期として，タスクスイッチに伴うスレッド実行終了時が挙げられる．または割り込みを利用して一定時間毎に処理することも可能である．

次にページへのアクセス有無の検出処理について説明する．アクセス有無を検出する方法として確実な方法は，プロセッサが残す記録を利用する方法である．この方法の利点は，ページアクセスの検出処理が不要なため記録を集計する以外の処理が不要な点である．

プロセッサによるページアクセス有無の記録先はページテーブルである．このページテーブルからアクセスページを得る方法を説明する．ページテーブルの各エントリはページの物理アドレスと管理情報を持つ．管理情報には参照されたことを示すフラグが存在する．よって各エントリの参照フラグを調べることでアクセ

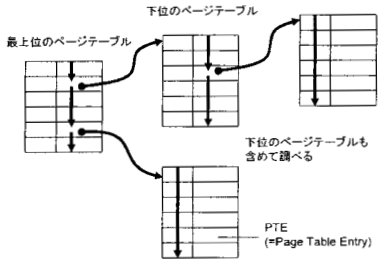

図 3　ページテーブルの階層構造

スページを特定可能である．しかしページテーブルは図 3 に示した通り階層構造により多くのエントリを持つため処理負荷が高い問題を抱える．そのためページテーブルのアクセスページ検出への利用は困難である．

プロセッサのアクセスページはページテーブル以外に TLB に記録される．TLB はページテーブルのキャッシュであるため，記録されたページはアクセスページである．我々は以前 TLB によるキャッシュ利用分布の推定法[8] を提案した．この方法では TLB に格納されたページアドレスを直接読み出すことでアクセスページを検出する．しかし TLB に記録されたアドレスを読み出すことが不可能なプロセッサが多く存在する．そのため TLB のアクセスページ検出への利用は使用可能な環境が制限される問題を抱える．

そこで本稿の提案ではページテーブル，TLB は使用せずにソフトウェア的にアクセスページを検出する．提案方式ではページアクセスを契機に発生する例外処理を用いてページアクセスを検出する．以下，具体的な手順を図 4 を用いて説明する．まずページアクセスの検出に利用する例外処理は，ページアクセス時に発生する Page Fault または TLB ミスの例外処理である．また例外をページの初回アクセス時に意図的に発生させるために事前に TLB の flush，及びページテーブルエントリの無効化を行う．ページアクセスが発生するとアドレス問い合わせ (1) の結果，Page Fault 例外 (2) が発生する．例外処理ルーチンにおいて，アクセス先ページをアクセスページに追加 (3) する．一覧への追加後は同一ページの検出は不要なため，アクセスページに追加されたページのエントリを有効化する (4)．これによりその後の例外発生を抑制する．

3.2　ページ参照頻度の推定方式

検出されたアクセスページに対して，各ページの参照頻度を推定する．アクセスページはアクセス頻度に関わらずアクセスされた全てのページを含むため，キャッシュの使用状況を推定するためには各ページのアクセス頻度についての情報が必要である．

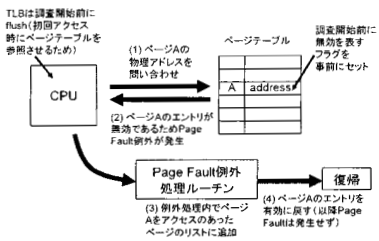

図 4　例外処理を利用したアクセスページ検出

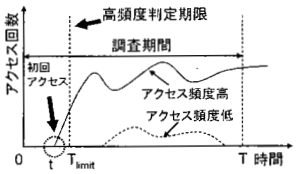

図 5　アクセス頻度と初回アクセス時間の関係

本項ではアクセスページの検出処理に続いて，ページ参照頻度を推定するために以下の 2 方式を提案する．

- 短期的なアクセス頻度推定方式
- 長期的なアクセスパターン推定方式

短期的とはアクセスページ及びアクセス頻度を調べ始めて 1 回分が終了するまでを調査期間と定義した場合，単一の調査期間内を表す．一方長期的とは単一の調査期間より長い，複数の調査期間を表す．またこれらの方式によるページ参照頻度推定処理を行う時期は，アクセスページの検出，集計処理と同時である．

3.2.1　短期的なアクセス頻度推定方式

短期的なアクセス頻度推定方式では，アクセス頻度の調査を開始してから初回アクセスまでの経過時間を基準にアクセス頻度の推定を行う．アクセス頻度と経過時間との関係は Cache Decay[5] の研究で用いられている．キャッシュエントリ毎の最終参照時刻を記録し，参照時刻が古いものを不使用と判断する．本提案方式ではアクセス頻度が高いページは初回アクセスまでの時間が短い確率が高くなる性質を用いている．

ページのアクセス頻度と初回アクセスまでの経過時間との関係を図 5 に示す．T はアクセスページ，及びアクセス頻度を調べるための時間単位を表す．

初回アクセス時間からアクセス頻度を判定するために，基準の時間を設定する必要がある．ここで初回アクセスが調査開始から T_{limit} 以内である場合をアクセス頻度が高いと判定する基準値とする．調査期間内のアクセス数が N 回のページのアクセス頻度が高いと判定される確率は，理論上 $1-(1-\frac{T_{limit}}{T})^N$ である．

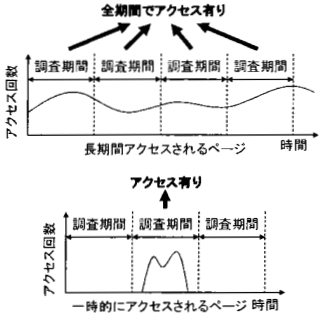

図 6　アクセスページの検出によるアクセスパターン推定

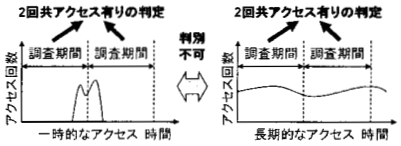

図 7　アクセスパターンの推定が困難な例

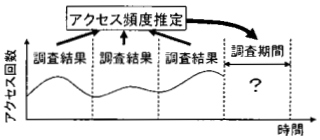

図 8　アクセス頻度，アクセスパターン推定の併用

N または T_{limit} が大きいほどアクセス頻度が高いと判定される確率が増大する．N はページアクセス数のため調整不可能だが，T_{limit} を調整することでアクセス頻度が高いと判定される確率の調整が可能である．

3.2.2　長期的なアクセスパターン推定方式

短期的なアクセス頻度推定方式は注目した調査期間内のアクセス頻度を推定する．しかし調査期間内におけるアクセスの時間変化を推定できない．一方ページアクセスの時間変化を調べることで，アクセス頻度推定精度の向上，アクセス頻度の予測を実現可能である．

そこで本稿では前節の推定方式と共に長期的なアクセスパターンの推定方式を提案する．アクセスの時間変化を推定するためには，過去のアクセス履歴が必要である．本推定方式では，アクセスページの過去複数回分の検出結果からアクセスパターンを推定する．アクセスパターンの推定方法を図 6 を用いて説明する．

例としてページに対して過去 4 回の調査期間全てでアクセスされている場合，図に示した通り 1 回の調査期間より長い期間の間継続的にアクセスされていることを表す．そのため定期的にアクセスされるページであると推定でき，次回の調査期間もアクセスされると予測できる．一方で過去 3 回の調査期間のうち中央の 2 回目のみアクセスされている場合，図に示した通り単一調査期間より長い期間に着目すると，一時的にアクセスされたページと推定できる．よって次回の調査期間にアクセスされる可能性は低いと予測できる．

実際にアクセスパターンの推定を行う場合，履歴に使用する調査結果数を決定する必要がある．履歴となる調査の回数を増やすことでアクセスパターンをより詳細に推定可能である．しかし同時にアクセスパターン推定の処理時間及び必要メモリ容量が増加するため，必要十分な精度を確保可能な回数の決定が必要である．

提案方式では事前のアクセスページ調査回数を 3 回分を履歴に使用してアクセスパターンを推定する．調査回数を 2 回としない理由を図 7 を用いて説明する．履歴が 2 回分である場合，図示された 2 回の調査期間中の継続的なアクセスと，1 回目と 2 回目の調査期間の境目付近で発生した一時的なアクセスとの区別が困難である．一方履歴が 3 回分以上の場合，継続的なアクセスが行われる場合は全調査期間でアクセスが検出される．しかし調査期間の境目付近で一時的にアクセスされた場合は，境目に接する 2 回の連続した調査期間以外ではアクセスが検出されない．よって調査回数が 3 回の場合は，2 回の場合より判定精度が高い．

3.2.3　2 種類の推定方式の併用による高精度化

前述のアクセス頻度，アクセスパターンの推定方式の併用により詳細なアクセスパターン推定が可能である．過去 3 回の調査期間におけるアクセス頻度の判定履歴から，次回調査期間における各ページのアクセス頻度の推定を行う (図 8)．これにより一方のみを用いた場合と比較してより詳細なアクセス頻度，アクセスパターンの記録が可能である．

まず 2 方式の併用によるアクセスパターンについて述べる．3 回の各調査期間におけるアクセス頻度判定結果は以下の 3 通りに分類される．

- 調査期間中にアクセス無し (パターン 0)
- 調査期間中のアクセス頻度低 (パターン 1)
- 調査期間中のアクセス頻度高 (パターン 2)

上記 3 種類の判定結果をそれぞれ 0,1,2 と表現する．調査は 3 回連続で行われるため，過去 3 回分のアクセスパターンは $3^3 = 27$ 通りに分類される．以下の記述において各パターンは，例えば過去 3 回古い方から順に調査結果が 0,2,1 である場合，"0-2-1" と表現する．

次にアクセスパターンとアクセス頻度の関係を図 9 の例を用いて説明する．まずアクセスパターン"2-2-2"の場合，つまり過去の全調査で初回アクセスが早く行われた場合を説明する．この場合，継続的に高頻度で

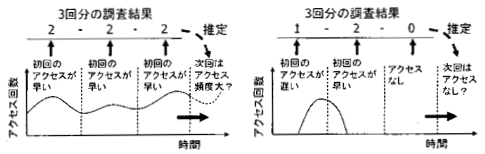

図 9　アクセスパターンからのアクセス頻度推定例

アクセスされていると推定される．そのため次回の調査期間でも高頻度でアクセスされると推定できる．

一方でページのアクセスパターン”1-2-0” の例では，過去 3 回調査期間のうち直近の調査期間ではアクセスされていない．そのため過去の調査期間内で一時的にアクセスされていたページであり，次回の調査期間についてもアクセスされないと推定できる．

アクセスパターンからページのアクセス頻度を予測するためには各アクセスパターンを持つページの実際のアクセス頻度を求める必要がある．その手段として，ベンチマークプログラムを用いて各アクセスパターンとアクセス頻度との関係を求める方法が挙げられる．

4. 提案方式の評価

提案方式の正当性，有効性をシミュレータで評価する．本章では提案方式の評価環境，評価結果を述べる．

4.1 評価環境

(1)　短期的なアクセス頻度推定

(2)　アクセス頻度とアクセスパターン推定の併用

上記 2 提案方式による推定の有効性を評価する．(1) ではアクセス頻度の推定結果と実際のページアクセス頻度の関係を評価する．(2) では前述の各アクセスパターンと該当ページのアクセス頻度の関係を評価する．

アクセス頻度の推定，及び実際の各ページアクセス頻度の計測はシミュレータに機能を追加して評価する．シミュレータは SimCore/Alpha Functional Simulator[6] Version 2.0r1 を使用する．ページサイズは 8KB とする．ベンチマークは SPEC CINT2000 から正常動作しない 186.crafty, 253.perlbmk を除いた 10 種類のテストを用いる．入力データは ref を用いる．

各テストにおいて直前の調査期間でのページのアクセス頻度，アクセスパターンの調査処理を 1 億命令毎に行う．同時に次の調査期間に関するページアクセス頻度の推定処理を行う．各テストの結果は 1 億命令毎に行われる提案方式適用結果の平均値とする．また各テストではアクセスパターンの推定が可能になる，実行開始から 3 億命令以降の結果を用いて評価を行う．

4.2 短期的なアクセス頻度推定方式の評価

短期的なアクセス頻度の推定方式について評価結果

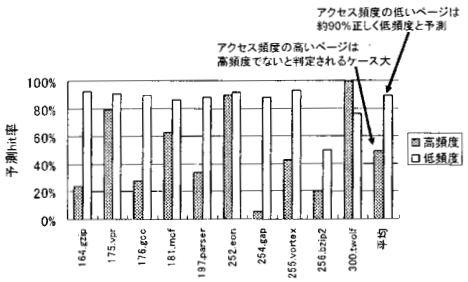

図 10　アクセス頻度の判定結果と実際のページアクセス頻度

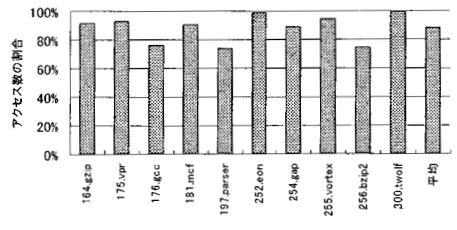

図 11　高アクセス頻度と判定されたページへのアクセス数の割合

と実際のページアクセス頻度から効果，正当性を評価する．本評価における固有の評価パラメータを説明する．まず調査期間は前述の通り $T = 100,000,000$ 命令分である．またアクセス頻度が高いと判定する，初回ページアクセス時間の閾値は $T_{limit} = 500,000$ 命令分である．実際のページアクセス頻度については，アクセス回数が $N \geq 1000$ の場合に高頻度と判定する．

まずアクセス頻度の判定結果と実際のページアクセス頻度との関係は図 10 に示した通りである．アクセス頻度の低いページでは平均で約 90%，アクセス頻度を低いと判定している．一方でアクセス頻度の高いページに関しては平均で約 50%のみがアクセス頻度が高いと判定されている．アクセス頻度の高いページの判定の hit 率が低い原因は，調査期間のうち一部の時間のみワーキングセットに含まれて一時的に高頻度でアクセスされるページが多いためと考えられる．

次にアクセス頻度が高くかつ提案方式でアクセス頻度が高いと判定したページへのアクセス数の，全アクセス数に占める割合を示す (図 11)．平均で約 88%のメモリアクセスが，提案方式によりアクセス頻度が高いと判定されたページに集中していることが分かる．

更に実アクセス数が多い，つまりアクセス数が N を上回るページのうち，提案方式によってアクセス頻度が低いと判定されたページへのアクセス数を 1 とした場合の，アクセス頻度が高いと判定された各ページ

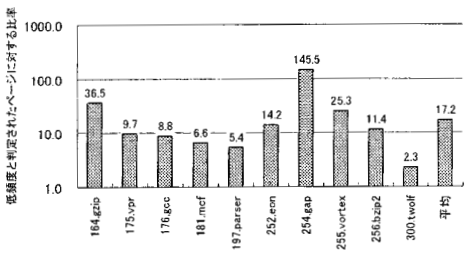

図 12　アクセス頻度が高いと判定されたページへのアクセス数比率

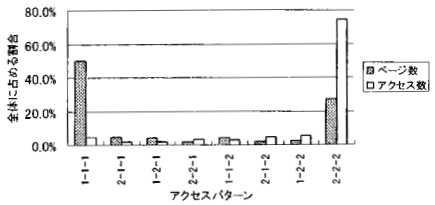

図 13　アクセスパターン別所属ページ数とページへのアクセス数

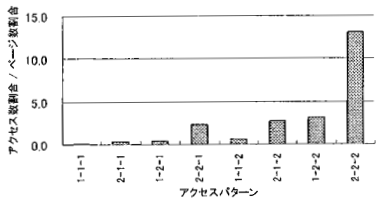

図 14　アクセスパターン別ページあたりのアクセス頻度

へのアクセス数比率を図 12 に示す．アクセス頻度が高いと判定されたページに対するアクセス数は平均で約 17.2 倍である．よってアクセス頻度が高いページほど実アクセス頻度が高いと判定されており，提案方式はアクセス頻度判定に効果を持つことを示している．

4.3　アクセス頻度，パターンの推定の併用

短期的なアクセス頻度推定方式，及び長期的なアクセスパターン推定方式の併用効果をアクセスパターンと実際のページアクセス頻度との関係から評価する．

まずアクセスパターンと実際のページアクセス頻度との関係は図 13 に示した通りである．例えばアクセスパターンが"2-2-2" であるページ数は平均で全体の約 27%，アクセス数は全体の約 74%である．一方でアクセスパターンが"1-1-1" であるページ数は平均で全体の約 50%，アクセス数は全体の 5%未満である．つまりページあたりのアクセス頻度が高いアクセスパターンと，低いアクセスパターンが存在する (図 14)．

本稿の提案方式を用いる目的であるキャッシュ使用状況の推定を行う場合，頻繁にアクセスされ，キャッシュされる可能性の高いページを集計する必要がある．前述の評価結果によりアクセス頻度の高いページは特定のアクセスパターンを持つことが示される．よって，短期的なアクセス頻度推定と，長期的なアクセスパターンの推定を併用する本稿の提案方式は，アクセス頻度判定に対して効果を持つことを示している．

5. ま と め

本稿では，各ページのアクセス頻度調査に基づくキャッシュ使用状況推定の有用性を述べた．我々は各ページのアクセス頻度を調べる方式として，調査開始から初回アクセスに要する時間によるページアクセス頻度の推定方式，アクセスページ調査の複数回の履歴によるアクセスパターン推定方式，及び前述 2 方式を併用したアクセスパターン推定方式を提案した．評価の結果，提案方式によるアクセス頻度の判定結果と実際のアクセス頻度の関連性から，提案方式のアクセス頻度の高いページの検出への有用性を確認した．

参 考 文 献

1) Brian K. Bray, William L. Lynch, M. J. Flynn: Page Allocation to Reduce Access Time of Physical Caches, Technical Report CSL-TR-90-454, Computer Systems Laboratory, Stanford Univ., 1990
2) R. E. Kessler, Mark D. Hill: Page Placement Algorithms for Large Real-Indexed Caches, ACM Transactions on Computer Systems, Vol. 10, No. 4, pp.338-359, 1992
3) B. Bershad, D. Lee, T. Romer, J. Chen: Avoiding Conflict Misses Dynamically in Large Direct-Mapped Caches, 6th ASPLOS, pp.158-170, 1994
4) Theodore H. Romer, Dennis Lee, Brian Bershad, J. Bradley Chen: Dynamic Page Mapping Policies for Cache Conflict Resolution on Standard Hardware, Symp. on Operating Systems Design and Implementation, pp.255-266, 1994
5) S. Kaxiras, Z. Hu, M. Martonosi: Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power, Proc. Annu. Int. Symp. on Computer Architecture, pp.240-251, 2001
6) Kenji Kise, Takahiro Katagiri, Hiroki Honda, Toshitsugu Yuba: The SimCore/Alpha Functional Simulator, Workshop on Computer Architecture Education (WCAE-2004) held in conjunction with the ISCA-31, 2004, http://www.arch.cs.titech.ac.jp/~kise/SimCore/index.htm
7) 小川 周吾, 平木 敬: プロセスの実行時情報を用いたスケジューラによる高速化手法, 情報処理学会論文誌, Vol. 46, No. SIG 12(ACS 11), pp.161-169, 2005
8) 小川 周吾, 平木 敬: TLB を用いた CPU キャッシュ利用分布の推定法, 情報処理学会研究報告, Vol.2006, No.88(2006-ARC-169), pp.85-90, 2006